Testing Like A Boss

Arkady Galyash

TosChart

May 3, 2014

Agenda

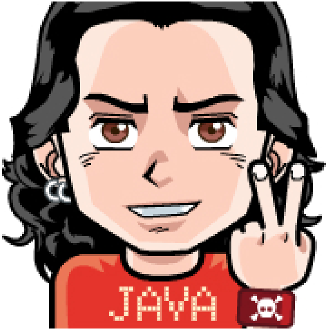
1. How regular JUnit test looks like (in time)?

2. Links

Intro

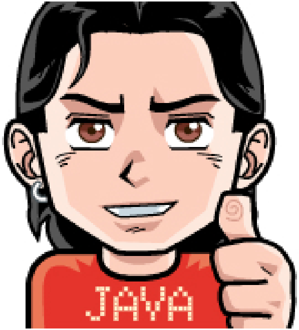
- John Doe The Programmer
- Java developer @ Moon Ms
- Since 2000
- Binary search algorithm

June 2000



Just graduated, "tests are for chicken"

May 2001



JUnit discovered

July 2006

User tries to search non-existant element

December 2007

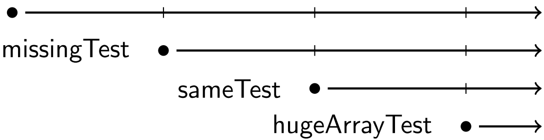
User tries to search in array with duplicates

October 2009

Integer overflow

Long way

2000 2001 2006 2007 2009



simpleTest

missingTest

sameTest

hugeArrayTest

xUnit

- Design tests example by example
- Test suites give us confidence that code works for the examples we thought of
- If we discover a test case, this test is useless right now, it may be useful only in future
- "Don't ask, don't tell"

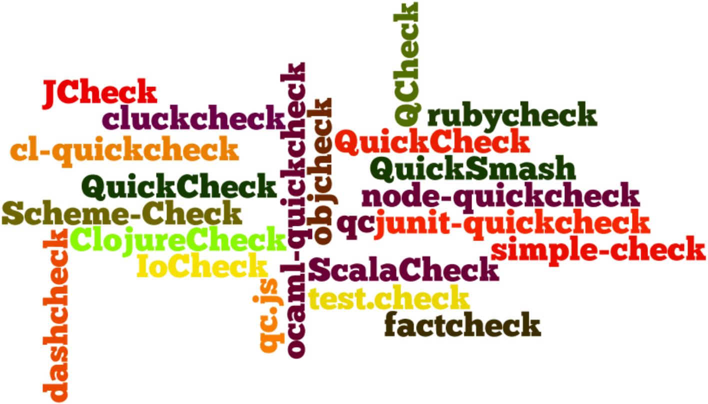
QuickCheck

- Another approach
- You specify actions and post-conditions
- Let computer generate input data
  (he will not forget about corner cases)

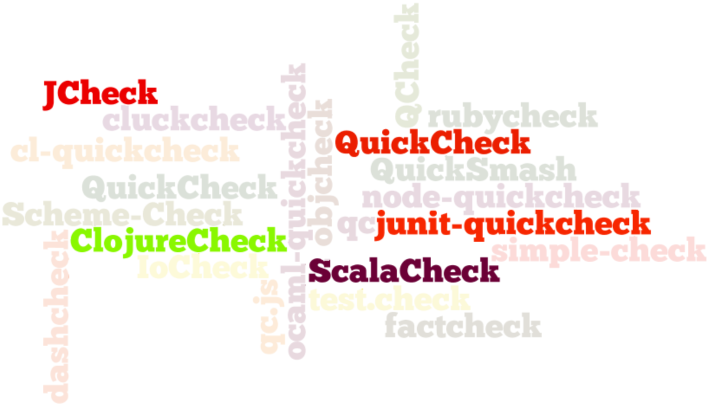
QuickCheck History

- Initially written for Haskell in 1999
- By Koen Claessen and John Hughes
- BSD-style License

QuickCheck implementations



QuickCheck implementations



junit-quickcheck

- Based on `org.junit.experimental.theories.*`
- Annotation-based
- On github of Paul Holser (@pholser)
- 0.3 version
- Found bug with generics during making presentation

JUnit Theory

- Experimental JUnit feature
- `@RunWith(Theories.class)`
- `@Theory`, `@DataPoint`, `@DataPoints`

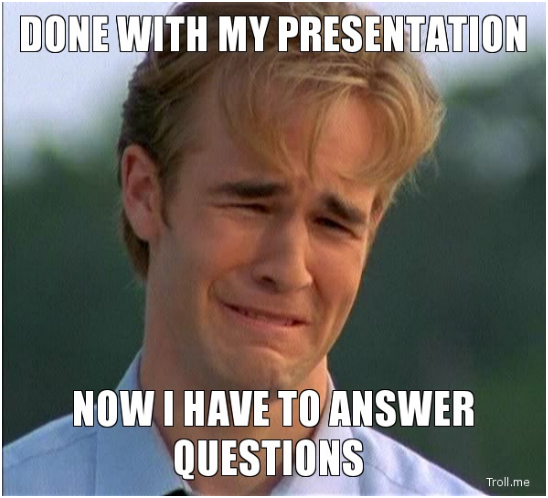
Demo

We Can Check It!
POST FEB. 15 TO FEB. 28
WAR PRODUCTION CO-ORDINATING COMMITTEE

Links

- Introduction to QuickCheck2

Thank you!


DONE WITH MY PRESENTATION
NOW I HAVE TO ANSWER QUESTIONS
Troll.me